APPENDIX B

Applicant's last of the DISCO document dated July 7, 2000 (4 pages)

Discovery of Web Services (DISCO)

Early Draft, 06 July 2000

Introduction

Web Services are described in one or more related documents using the SOAP Contract Language (SCL). This document proposes how SCL documents can be retrieved for a resource or collection of resources, e.g. how information about a service is **discovered**. Note that this discovery algorithm it is not limited to discovering information about web services; it can be used to find metadata about any resource.

There are other existing mechanisms for discovery (DAV, RDF, ...) that are just as valid for discovering services. This document does not attempt to be an exhaustive list but instead proposes two lightweight mechanisms that are likely to work with existing infrastructure.

Notational Conventions

The keywords "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC-2119 [2].

The namespace prefixes "disco", "scl", "schema" and "soap" used in this document are associated with the DISCO namespaces "http://schemas.xmlsoap.org/disco/", "http://schemas.xmlsoap.org/disco/scl/", "http://schemas.xmlsoap.org/disco/schema/" and "http://schemas.xmlsoap.org/disco/soap/".

Status

This draft represents the current thinking with regard to discovery of services within Microsoft. It is published by Microsoft for the purpose of informing interested parties of the current state of that thinking, not as a finished proposal or commitment to implement. Microsoft will not allow early implementation to constrain its ability to make changes to this specification prior to final release.

Table of Contents

1. Discovery Algorithm
2. Discovery Document Format
3. SCL and discovery
   3.1 contractRef
4. Schemas and discovery
   4.1 schemaRef
5. SOAP and discovery
   5.1 soap

1. Discovery Algorithm

The discovery algorithm is based on the existence of a discovery document format (an XML grammar). A discovery document is the result of performing discovery on a resource. A discovery document contains links to other resources describing or otherwise related to the resource (Service Contracts, Schemas, etc). A discovery document may refer to related discovery documents. The details of the discovery document format are covered in section 2.

The following algorithm allows a client to retrieve a discovery document given the URL of a resource. The algorithm is independent of what the URL represents. For example, it could be the URL of the site, the URL of a folder within the site, the URL of a particular service within the site, etc.

To obtain the discovery document ***D*** associated with a URL ***U***:

- Perform an HTTP GET on ***U*** producing response ***R***
- If the HTTP content-type of ***R*** is "text/xml":
  - If ***R*** contains a discovery document, it is the discovery document D
  - Otherwise, if there is an xml-stylesheet processing instruction within the content of ***R*** with at least the following attribute values:
    - type='text/xml'
    - alternate='yes'
    - href='***U2***'

    Then
    - Perform an HTTP GET on ***U2*** producing response ***R2***
    - If ***R2*** contains a discovery document, it is the discovery document ***D***
    - Otherwise, there is no discovery document ***D*** available for ***U***
  - Otherwise, there is no discovery document ***D*** available for ***U***
- If the HTTP content-type of ***R*** is "text/html":
  - If there is an occurrence of a LINK tag within the content of ***R*** with at least the following attribute values:
    - type='text/xml'
    - rel='alternate'
    - href='***U2***'

    Then
    - Perform an HTTP GET on ***U2*** producing response ***R2***
    - If ***R2*** contains a discovery document, it is the discovery document ***D***
    - Otherwise, there is no discovery document ***D*** available
- Otherwise, there is no document ***D*** available for ***U***

The following MUST be true in order for an HTTP response to be considered a discovery document:

- The content-type of the HTTP response is "text/xml".
- The name of the root XML element conforms to the discovery document format.
- Either the XML namespace is not specified, or it matches that of the discovery document format.

See http://www.w3.org/TR/xml-stylesheet/#XML for more information on the **xml-stylesheet** processing instruction.
See http://www.w3.org/TR/html401/struct/links.html#edef-LINK for more information on HTML's LINK element.

2. Discovery Document Format

The discovery document format is a container for elements that typically contain links (URLs) to resources that provide discovery information. If the URL's are relative, they are assumed to be relative to the location of the discovery document.

The following example discovery document refers to another discovery document, a Service Contract document, a schema document, and HTML based documentation for the service.

```
<disco:discovery>
  <disco:discoveryRef ref='folder/discovery'/>

  <-- elements from other namespaces -->
</disco:discovery>
```

The schema for the discovery document is neutral as to what types of things are being discovered. It relies on elements from other namespaces to provide the actual linking mechanisms specific to a resource type or category.

The only linking element defined intrinsically is **discoveryRef**, which refers to another discovery document. The value of the **ref** attribute is the URL of the discovery document.

Discovery documents may include the XML namespace "http://schemas.xmlsoap.org/disco/" for the **discovery** tag. Programs processing these documents **MUST** read and understand the namespace and **SHOULD** reject documents where the namespace is present and different.

Programs processing discovery documents **SHOULD** ignore any unrecognized elements under the **discovery** tag to allow extensibility of the format.

A discovery document **MAY** have an XML style sheet processing instruction.

3. SCL and discovery

The following elements are defined by the XML namespace "http://schemas.xmlsoap.org/disco/scl/" for use within discovery documents.

3.1 contractRef

The **contractRef** element refers to service defined using the SOAP Contract Language (SCL). The value of the **ref** attribute is the URL of the document. The value of the optional **docRef** attribute is the URL to human readable documentation for the service.

The following example refers to two service contracts:

```
<disco:discovery>
  <scl:contractRef ref='my1.sdl'>
  <scl:contractRef ref='my2.sdl' docRef='my.htm'>
</disco:discovery>
```

4. Schemas and discovery

The following elements are defined by the XML namespace "http://schemas.xmlsoap.org/disco/schema/" for use within discovery documents.

4.1 schemaRef

The **schemaRef** element refers to an XML schema. The value of the **ref** attribute is the URL of the schema. The optional **targetNamespace** attribute indicates the namespace of the schema.

The following example refers to two schemas:

```
<disco:discovery>
  <schema:schemaRef ref=‘my1.xsd’ targetNamespace=‘http://my.org/my1.xsd’/>
  <schema:schemaRef ref=‘my2.xsd’/>
</disco:discovery>
```

5. SOAP and discovery

The following elements are defined by the XML namespace “http://schemas.xmlsoap.org/disco/soap/” for use within discovery documents.

5.1 soap

The **soap** element specifies the location of a soap service with a particular soap **binding** that has been defined in a SCL document. The **binding** attribute contains a QName value. The URI portion of the QName refers to the targetNamespace of a SCL document. The name portion of the QName refers to a **binding** within that SCL document. The value of the **address** attribute is the URL of the soap service.

The following example provides information about two web services:

```
<disco:discovery>
  <soap:soap binding=‘my1:binding1’ address=‘my1’>
  <soap:soap binding=‘my1:binding2’ address=‘my1’>
  <soap:soap binding=‘my1:binding1’ address=‘my2’>
</disco:discovery>
```

Although this information is available by reading a SCL document, this lightweight mechanism allows easier processing by clients attempting to find the location of a service with a known binding.